



# Antipattern Einführung

---

von Lars Münzberg



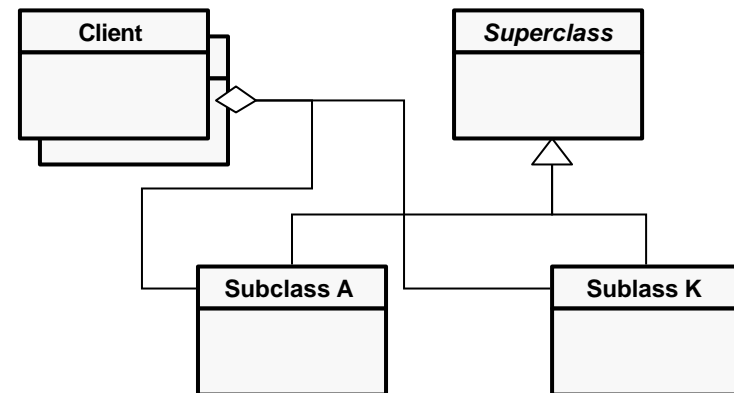
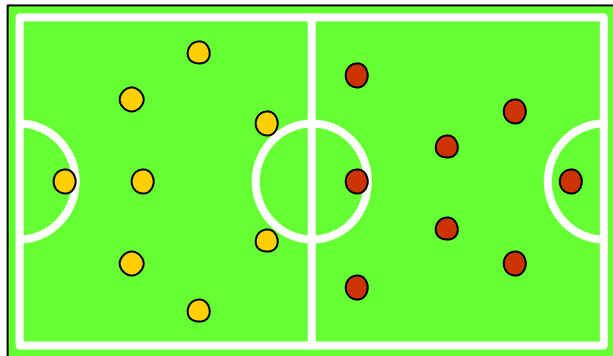
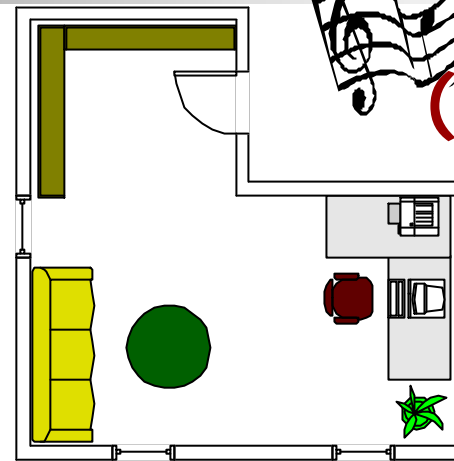
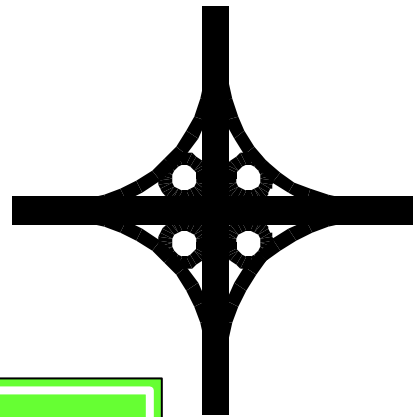
# Inhalt

---

1. Einführung
2. Von Pattern zu Antipattern
3. Antipattern Referenzmodell
4. Quellen

# 1. Einführung

Muster ?





# 1. Einführung

---

Pattern?

**... are about what works\***

- nicht Neues, sondern Bewährtes beschreiben
  - in bestimmter Form
- ⇒ „Experten“wissen festhalten

\* B. Foote, PLOPD 3



# 1. Einführung

---

- **Aktueller Stand:**
  - **Pattern Almanac 2000 (Linda Rising):**
    - > 1000 Muster
    - ca. 80 ‚Sammlungen‘ (Systeme bzw. Sprachen)
    - 68 Kategorien, z.B.  
Accounting, Analysis, Behavioral, Client-Server,  
Concurrent Systems, Security, Testing, Visualisation &  
Monitoring, ...



# 1. Einführung

---

- Software wird heute immer komplexer, also möglichst Komponenten wiederverwenden
- Studie von J. Johnson „Creating Chaos“ (1995)
  - 5/6 aller Projekte sind Mißerfolge
  - 1/3 aller Softwareprojekte werden gestoppt
  - Rest zumeist mit doppeltem Budget oder Zeit realisiert



# 1. Einführung

---

- Wie verbessert man dies?
  - Antipattern, überall zu finden
  - 4 Variablen determinieren gleichzeitig die Kontrolle von SW-Projekten:
    - Kosten
    - Qualität
    - Zeit
    - Scope
  - man lässt den Kunden drei auswählen



## 2. Von Pattern zu Antipattern

---

- Pattern:
  - Beschreibung einer angestrebten „guten“ Lösung für ein typisches Problem
- Antipattern:
  - Beschreibung einer existierenden „schlechten“ Lösung für ein typisches Problem; also verursacht mehr Probleme als es löst
  - Identifikation von Problemstellen
  - entsteht auch durch falsche Anwendung von Pattern
  - weniger konkret und technisch detailliert





## 2. Von Pattern zu Antipattern

---

- Warum untersuchen wir dieses Konzept?
  - in der Praxis oft mehr schlechte Lösungen als gute
  - reichhaltiges Feld für Verbesserungen
  - zeigt die Bedeutung des Verständnisses über Kontext und Konsequenz von Entwurfsentscheidungen

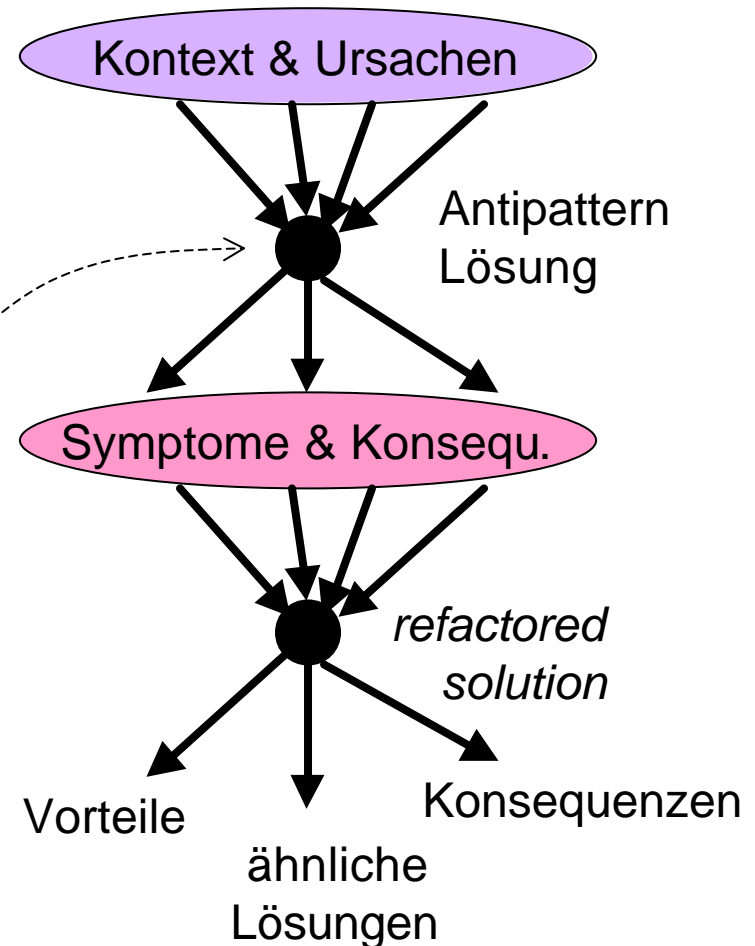
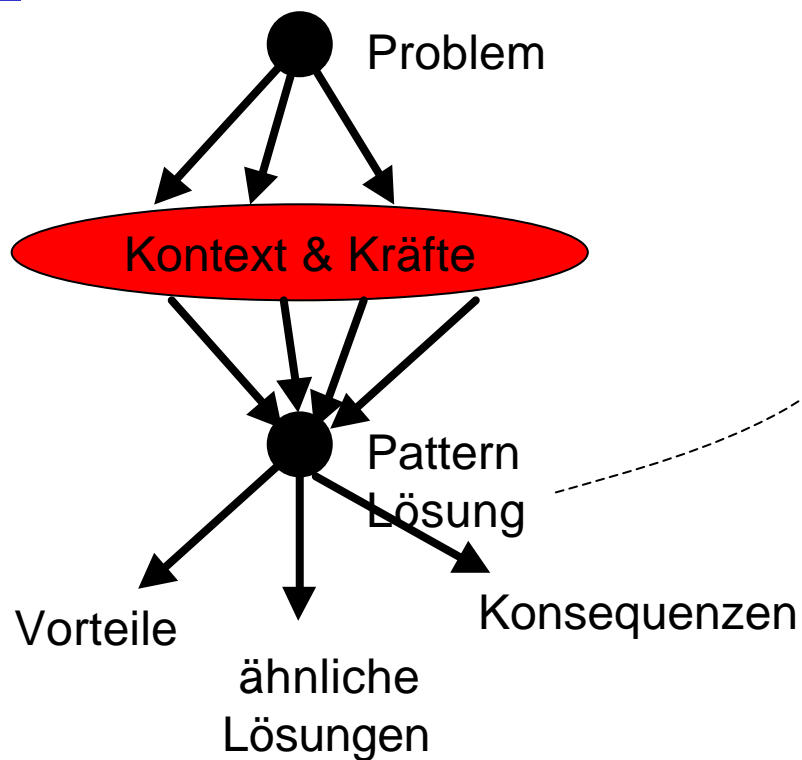


## 2. Von Pattern zu Antipattern

---

- Nutzen:
  - Wissen zum Verhindern, Entdecken und Verbessern von negativen Lösungen
  - Hilfestellung und praxiserprobte Lösungsvorschläge geben
  - Brücken zwischen architektonischen Konzepten und realen Implementierungen bilden
  - Verallgemeinerung, Formalisierung von Verbesserungen
  - Ausgangspunkt von Veränderungen

## 2. Von Pattern zu Antipattern





## 2. Von Pattern zu Antipattern

---

- Beispiele:

- Spaghetticode
- der Klumpen (*the blob*):
  - Riesenhauptobjekt, das alles tut
  - weitere Objekte nur Datenhalter



## 3. Antipattern Referenzmodell

---

- allgemeine Modelvorstellung des Problembereichs Softwareentwicklung
- Definiert alle gemeinsamen Konzepte, die den Antipattern und ihrer Darstellungsform zugrundeliegen

Ziel: Verkürzung der Beschreibung



## 3. Antipattern Referenzmodell

---

- Inhalt:

- Root Causes

- grundlegende Fehler in der Softwareentwicklung
- problembereichs- bzw. projektunabhängig

- Primal Forces

- Ansprüche bzw. Anforderungen an eine Architektur
- horizontale (problemereichsunabhängige) Urkräfte, also grundlegende Überlegungen für den Erfolg eines Softwareprojekts

- Software Design-Level Modell (SDLM)



## 3. Antipattern Referenzmodell

---

- Root Causes:
  - Haste
  - Apathy
  - Narrow-mindedness
  - Sloth
  - Avarice
  - Ignorance
  - Pride



## 3. Antipattern Referenzmodell

---

- Primal Forces:
  - Management of functionality
  - Management of performance
  - Management of complexity
  - Management of change
  - Management of IT resources
  - Management of technology transfer

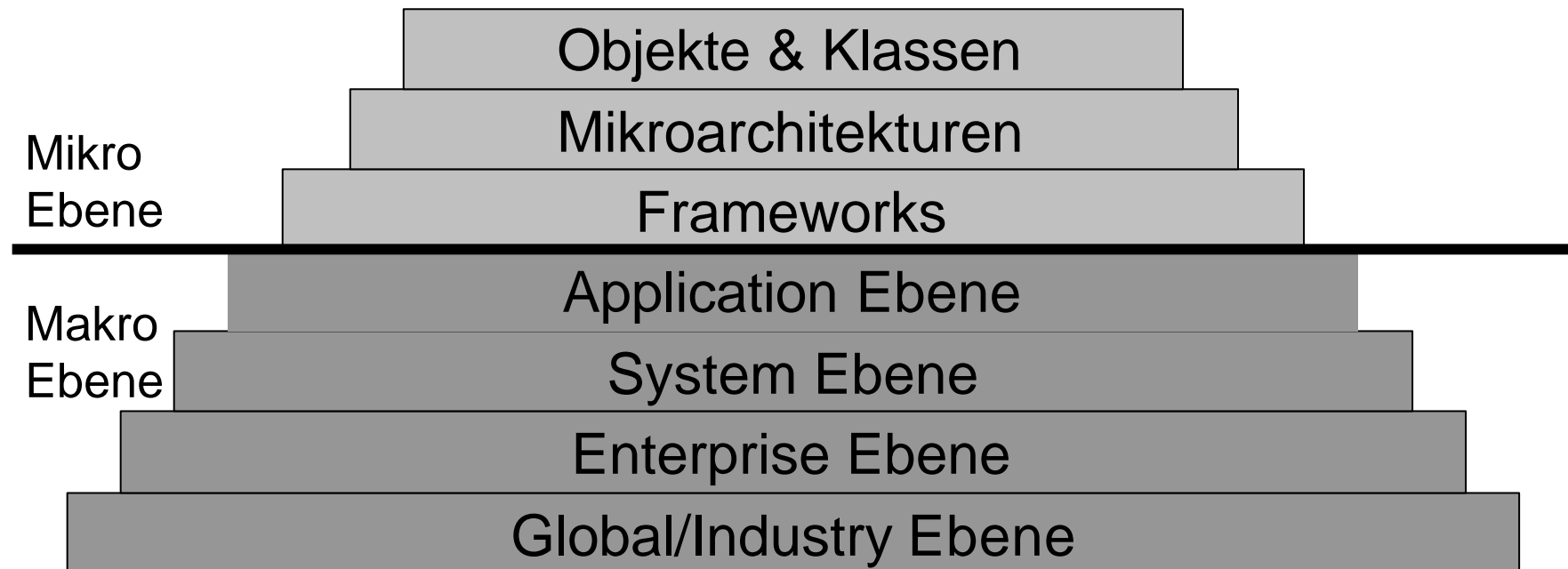




# 3. Antipattern Referenzmodell

---

## SDLM





## 3. Antipattern Referenzmodell

---

- Schablone:
  - Name
  - also known as
  - zugehörige Ebene des SDLM, Einfluss darauf
  - Root Causes
  - Anekdote
  - Charakteristika
  - Symptome und Konsequenzen
  - Refactored Solutions
  - Beispiel
  - verwandte Antipattern
  - Pattern die bei falscher Anwendung zu diesem Antipattern führen



## 10. Quellen

---

- W. J. Brown, J. Wiley: „AntiPatterns“
- Gamma, Helm, Johnson, Vlissides: „Entwurfsmuster“
- Fraunhofer Institut: Informations- und Datenverarbeitung
- Vlissides: „Entwurfsmuster anwenden“



# Iterator-Pattern

---

Vielen Dank für Ihre  
Aufmerksamkeit.